

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Bleizeffer et al.	§
	§ Group Art Unit: 2176
Serial No. 10/712,467	§
	§ Examiner: Dasgupta, Soumya
Filed: November 13, 2003	§
	§ Confirmation No.: 9433
For: Inline Representation of Steps in a	§
Multi-Stepped Process	§

36736

PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on July 25, 2008.

A fee of \$540.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0461. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0461. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0461.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

This appeal has no related proceedings or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

The claims in the application are: 1-24

B. STATUS OF ALL THE CLAIMS IN APPLICATION

Claims canceled: None

Claims withdrawn from consideration but not canceled: None

Claims pending: 1-24

Claims allowed: None

Claims rejected: 1-24

Claims objected to: None

C. CLAIMS ON APPEAL

The claims on appeal are: 1-24

STATUS OF AMENDMENTS

No Amendment after final rejection was filed for this case.

SUMMARY OF CLAIMED SUBJECT MATTER

A. CLAIM 1 - INDEPENDENT

The subject matter of claim 1 is directed to a method for presenting a step of a task, wherein the task includes a series of steps to be performed (Specification p. 11, ll. 26-29). The method comprises identifying a current step within the series of steps (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 718). The method comprises retrieving a step component for the current step (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 720). The method also comprises presenting the current step inline within the series of steps such that the step component is presented in context within the series of steps (Specification p. 19, ll. 7-10; Fig. 7, reference numeral 710).

B. CLAIM 11 - INDEPENDENT

The subject matter of claim 11 is directed to an apparatus for presenting a step of a task, wherein the task includes a series of steps to be performed (Specification p. 11, ll. 26-29). The apparatus comprises means (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 718) for identifying a current step within the series of steps (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 718). The apparatus comprises means (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 720) for retrieving a step component for the current step (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 720). The apparatus comprises means (Specification p. 19, ll. 7-10; Fig. 7, reference numeral 710) for presenting the current step inline within the series of steps such that the step component is presented in context within the series of steps (Specification p. 19, ll. 7-10; Fig. 7, reference numeral 710).

C. CLAIM 12 - DEPENDENT

The subject matter of claim 12 is directed to the apparatus of claim 11, wherein the means (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 718) for identifying a current step within the series of steps includes means (Specification p. 18, ll. 6-7; Fig. 7, reference numeral 702) for receiving a request from a client (Specification p. 18, ll. 6-7; Fig. 7, reference numeral 702).

D. CLAIM 15 - DEPENDENT

The subject matter of claim 15 is directed to the apparatus of claim 12, wherein the means (Specification p. 19, ll. 7-10; Fig. 7, reference numeral 710) for presenting the current step inline within the series of steps includes means (Specification p. 18, ll. 13-17; Fig. 7, reference numeral 708) for generating a response page using a JAVA Server Page (Specification p. 18, ll. 13-17; Fig. 7, reference numeral 708).

E. CLAIM 16 - DEPENDENT

The subject matter of claim 16 is directed to the apparatus of claim 15, wherein the means (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 720) for retrieving a step component for the current step includes means (Specification p. 19, ll. 19-22; Fig. 7, reference numeral 720) for retrieving the step component using a Tiles framework (Specification p. 19, ll. 19-22; Fig. 7, reference numeral 720).

F. CLAIM 17 - DEPENDENT

The subject matter of claim 17 is directed to the apparatus of claim 15, wherein the means (Specification p. 18, ll. 13-17; Fig. 7, reference numeral 708) for generating a response page includes means (Specification p. 19, ll. 19-22; Fig. 7, reference numeral 720) for building the response page using a Struts framework (Specification p. 19, ll. 19-22; Fig. 7, reference numeral 720).

G. CLAIM 19 - DEPENDENT

The subject matter of claim 19 is directed to the apparatus of claim 18, further comprising: means (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 718) for identifying a current task within the plurality of tasks (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 718); means (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 720) for retrieving a task tile for the current task (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 720); and means (Specification p. 19, ll. 7-10; Fig. 7, reference numeral 710) for presenting the task tile as a series of steps (Specification p. 19, ll. 7-10; Fig. 7, reference numeral 710).

H. CLAIM 20 - DEPENDENT

The subject matter of claim 20 is directed to the apparatus of claim 15, wherein the means (Specification p. 19, ll. 7-10; Fig. 7, reference numeral 710) for presenting the current step inline within the series of steps further includes means (Specification p. 19, ll. 7-10; Fig. 7, reference numeral 710) for sending the response page to the client (Specification p. 19, ll. 7-10; Fig. 7, reference numeral 710).

I. CLAIM 21 - INDEPENDENT

The subject matter of claim 21 is directed to a server for presenting a step of a task, wherein the task includes a series of steps to be performed (Specification p. 11, ll. 26-29). The method comprises a controller (Specification p. 8, ll. 15-17, Fig. 2, reference numeral 208), wherein the controller receives a request from a client (Specification p. 18, ll. 6-7; Fig. 7, reference numeral 702), wherein the request identifies a current step within the series of steps (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 718), and wherein the controller retrieves a step component for the current step (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 720). The method comprises a JAVA Server Page (Specification page), wherein the JAVA Server Page builds a response page (Specification page) and wherein the response page presents the current step inline within the series of steps such that the step component is presented in context within the series of steps (Specification page).

J. CLAIM 24 - INDEPENDENT

The subject matter of claim 24 is directed to a computer program product (Specification p. 20, l. 3). The computer program product comprises a recordable-type medium having instructions for presenting a step of a task (Specification p. 20, l. 3), wherein the task includes a series of steps to be performed (Specification p. 11, ll. 26-29). The computer program product comprises instructions for identifying a current step within the series of steps (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 718). The computer program product comprises instructions for retrieving a step component for the current step (Specification p. 19, ll. 4-6; Fig. 7, reference numeral 720). The computer program product comprises instructions for presenting the current step inline within the series of steps such that the step component is presented in context within the series of steps (Specification p. 19, ll. 7-10; Fig. 7, reference numeral 710).

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to review on appeal are as follows:

A. GROUND OF REJECTION 1

Whether *Keane et al.*, Managing Print Jobs, U.S. Patent No. 6,650, 433 (November 18, 2003) (hereinafter “*Keane*”) fails to anticipate claims 1-4, 11-14 and 24.

B. GROUND OF REJECTION 2

Whether the Examiner failed to state a *prima facie* obviousness rejection against claims 5, 6, 8-10, 15, 16, 18-21 and 23 under 35 U.S.C. § 103 as obvious over *Keane* in view of *Hind et al.*, Achieving Application-Specific Document Content by Transcoding using Java Server Pages, U.S. Patent No. 6,715,129 (March 30, 2004) (hereinafter “*Hind*”).

C. GROUND OF REJECTION 3

Whether the Examiner failed to state a *prima facie* obviousness rejection against claims 7, 17, and 12 under 35 U.S.C. § 103 as obvious over *Keane* view of *Hind* and in further view of *Scheinblum*, Make Your Applications Strut, <http://articles.techrepublic.com.com/5100-22-1027640.html> (March 5, 2002) (hereinafter “*Scheinblum*”).

ARGUMENT

A. GROUND OF REJECTION 1 (Claims 1-4, 11-14 and 24)

A.1 Response to Rejection

The first ground of rejection is the erroneous assertion that *Keane* anticipates claims 1-4, 11-14 and 24. This rejection is respectfully traversed. The examiner states that:

Keane teaches a method for presenting a step of a task, wherein the task includes a series of steps to be performed, the method comprising: identifying a current step within the series of steps; (Fig 4 a-c → Keane teaches a system with a graphical user interface with a process with steps.. The current step is a highlighted step.)

retrieving a step component for the current step; (Fig 4 a-c → Keane teaches a system with a graphical user interface with a process with steps. The current step is a highlighted step. The user can input the settings of the current step.)

and presenting the current step inline within the series of steps such that the step component is presented in context within the series of steps. (Fig 4 a-c → Keane teaches a system with a graphical user interface with a process with steps. The current step is a highlighted step within a series of other steps.)

Final Office Action of April 25, 2008, p. 3 (emphasis in original).

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). In this case, each and every feature of the presently claimed invention is not identically shown in the cited reference, arranged as they are in the claims.

Claim 1 is as follows:

1. (Original) A method for presenting a step of a task, wherein the task includes a series of steps to be performed, the method comprising:
identifying a current step within the series of steps;

retrieving a step component for the current step; and
presenting the current step inline within the series of steps such that
the step component is presented in context within the series of steps.

Keane does not anticipate claim 1 because *Keane* does not teach the features of, “retrieving a step component for the current step,” or “presenting the current step inline within the series of steps such that the step component is presented in context within the series of steps,” as required in claim 1. The examiner incorrectly asserts otherwise, citing the following portions of *Keane*:

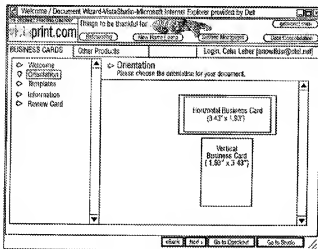


FIG. 4A

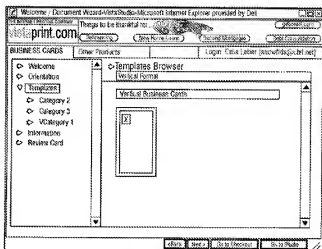


FIG. 4B

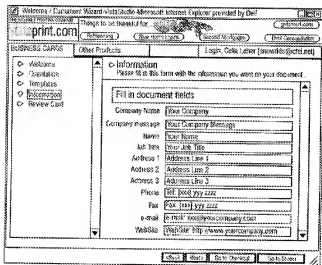


FIG. 4C

buttons, giving the customer a feeling of familiarity and user-friendliness. In the Design Wizard, the customer selects the item that the customer wishes to design (e.g., business cards or other items, in FIGS. 4-4E). For business card design, the Design Wizard includes a Welcome screen (FIG. 4), an Orientation screen (FIG. 4A) that allows the customer to choose between horizontal and vertical cards, a Template Browser screen (FIG. 4B) that allows the customer to choose between a variety of different design templates (not shown), an Information screen (FIG. 4C) at which the customer fills in a number of fields to complete the selected design template with the customer's information, and Review screens (FIGS. 4D and 4E) that allow the customer to review the front and back of the resulting business card. After reviewing the card, the customer can decide to (a) go back and edit the card, (b) go to the Checkout (the Purchase Wizard described below), or (c) go to the Design Studio to perform more complicated design functions (e.g., changing fonts and color schemes).

Keane, col. 11, ll. 18-43.

The cited portions of *Keane* teach a computer-implemented process that enables users at client workstations to generate business cards. In context, *Keane*'s claims are directed towards printing batches of business cards generated using the methods shown in figures 4A through 4O. *Keane*, claim 1, col. 22, ll. 2-18. Specifically, figure 4A shows, in the left hand portion of the screen, a list of steps taken in the process of laying out a business card to be printed; namely, "welcome," "orientation," "templates," "information," and "review card." Applicants assume, *arguendo only*, that these steps are equivalent to the claimed "series of steps" of "a task," as provided in claim 1. Next, figure 4B of *Keane* shows a submenu of items under the "templates" tab; namely, *Keane* shows a series of categories. Applicants assume, *arguendo only*, that

The portion of *Keane* describing these figures is as follows:

FIGS. 4-4O show webpages from a website studio used in one implementation of the invention. To begin the design process, the customer first navigates from a home page (not shown), to the Design Wizard (FIGS. 4-4E). The Design Wizard is configured to appear to the customer like a standard

Windows.RTM. Wizard application, e.g., with "back," "next" and "finish"

“templates” in *Keane* is equivalent to a “select template step” in the “series of steps to lay out a business card.” Next, figure 4C of *Keane* shows the details of the “information” step shown in figure 4A; namely, the document fields that should be filled in for the business card.

However, none of these features or descriptive text teach the above-identified features of claim 1. Applicants address these features one at a time. First, *Keane* does not teach, “retrieving a step component for the current step.” With respect to this feature, the examiner states, “Keane teaches a system with a graphical user interface with a process with steps. The current step is a highlighted step. The user can input the settings of the current step.”

However, the examiner has ignored the language of the claim. The claim requires, “retrieving a step component for the current step.” *Keane* does not teach *retrieving* step components. In fact, *Keane* does not teach “step components” at all. In figure 4B of *Keane*, cited by the examiner, the “categories” under the “templates” step are not step components, but rather categories of templates that the user can browse. For example, a user can view any one of the categories shown, at least at any one time. However, the categories are not part of the “templates” step, just selections of templates from which the user can choose. Hence, *Keane* does not teach “step components,” as required in claim 1.

Furthermore, as stated above, *Keane* does not teach *retrieving* such step components. Thus, *Keane* does not teach, “retrieving a step component for the current step,” as in claim 1. Accordingly, *Keane* does not anticipate this grouping of claims.

Additionally, *Keane* does not teach, “presenting the current step inline within the series of steps *such that the step component is presented in context within* the series of steps,” as in claim 1. As described above, *Keane* does not teach step components. Thus, *Keane* does not teach that the step components are presented in context within the series of steps, as in claim 1. Hence, again, *Keane* does not anticipate this grouping of claims.

A.2. Refutation of Examiner’s Response

In response to the facts established above, the Examiner asserts that:

Claims 1-4, 11-14, and 24 are Rejected Under 35 USC -102 (b):

Applicant’s arguments filed for Claims 1-4, 11-14, and 24 have been fully considered but they are not persuasive. The applicant argues that Keane does not disclose “retrieving a step component for the current step” or

"presenting the current step inline within the series of steps such that the step component is presented in the context within the series of steps."

The examiner disagrees.

Keane discloses "retrieving a step component for the current step" in that the process portrays the appropriate GUI components for the step selected. (Note: Fig 7 of the application discloses a flow chart for the algorithm of the proposed invention which has Item 720 as "Retrieve Step Tile for Current Step.") In Fig 4A and 48 of Keane, when the user selects "Orientation" icon in the left frame, the right frame discloses components of a GUI that portrays Horizontal and Vertical orientation of the business card (Fig 4A); when the user selects the "Template" icon in the left frame, the right frame discloses components of a GUI that portrays the templates for the business card (Fig 48). Keane also discloses "presenting the current step inline within the series of steps" in that in Figs 4A and 48, the "Orientation" and the "Template" icon is in line and in series (2nd and 3rd on the list respectively) with the icons that portray a process (see left frame). The "step components" are in the right frame, displaying the GUI components of the selected step that are showing segmented parts of a process.

Final Office Action of April 25, 2008, pp. 14-15.

The examiner is mistaken. None of these features or descriptive text teach the above-identified features of claim 1. *Keane* does not teach, "retrieving a step component for the current step." With respect to this feature, the examiner states, "In Fig 4A and 48 of Keane, when the user selects "Orientation" icon in the left frame, the right frame discloses components of a GUI that portrays Horizontal and Vertical orientation of the business card (Fig 4A); when the user selects the "Template" icon in the left frame, the right frame discloses components of a GUI that portrays the templates for the business card (Fig 48)."

Rather, the claim requires, "retrieving a *step component* for the current step." *Keane* does not teach *retrieving* step components. In fact, *Keane* does not teach "step components" at all. In figure 4B of *Keane*, cited by the examiner, the "categories" under the "templates" step are not step components, but rather categories of templates that the user can browse. For example, a user can view any one of the categories shown, at least at any one time. However, the categories are not part of the "templates" step, just selections of templates from which the user can choose. Hence, *Keane* does not teach "step components," as required in claim 1.

Furthermore, as stated above, *Keane* does not teach *retrieving* such step components. Thus, *Keane* does not teach, “retrieving a step component for the current step,” as in claim 1. Accordingly, *Keane* does not anticipate this grouping of claims.

Additionally, *Keane* does not teach, “presenting the current step inline within the series of steps *such that the step component is presented in context within the series of steps*,” as in claim 1. As described above, *Keane* does not teach step components. Thus, *Keane* does not teach that the step components are presented in context within the series of steps, as in claim 1. However, with respect to this feature, the examiner states, “Keane also discloses “presenting the current step inline within the series of steps such that the step component is presented in the context within the series of steps” in that in Figs 4A and 48, the “Orientation” and the “Template” icon is in line and in series (2nd and 3rd on the list respectively) with the icons that portray a process (see left frame). The “step components” are in the right frame, displaying the GUI components of the selected step that are showing segmented parts of a process.” This feature involves presenting a step component. *Keane* does not teach step components. Hence, again, *Keane* does not anticipate this grouping of claims.

B. GROUND OF REJECTION 2 (Claims 5, 6, 8-10, 15, 16, 18-21 and 23)

B.1. Response to Rejection

The second ground of rejection is the erroneous assertion that claims 5, 6, 8-10, 15, 16, 18-21 and 23 are obvious under 35 U.S.C. § 103 in view of *Keane* and *Hind*. This rejection is respectfully traversed. With respect to claim 5, the examiner states that:

Keane teaches the limitations of claim 1.

Keane discloses *a method presenting current step inline with a series of steps* (Fig 4 a-c → Keane teaches a system with a graphical user interface with a process with steps. The current step is a highlighted step within a series of other steps.)

Keane also discloses *Java*. (Col 13, lines 1-5 → Keane discloses a system which utilizes a Javascript.)

Keane does not appear to explicitly disclose *a method using a Java Server Page*.

However, Hind discloses *a method presenting a Java Server Page*.
(Abstract → Hind discloses a system that Java Server Pages.)

Keane and Hind are both analogous art because they are from the same field of endeavor of graphical user interface applications using Java.

At they time of the invention, it would have been obvious to one of ordinary skill in the art, having the teachings of Keane and Hind before him or her, to modify the GUI that identifies a current step within the series of steps to include communication with a client and server because it allows tasks to be listed and oriented in a network system with an universal web based language like Java, as disclosed by Keane, with a Java Server Page, as disclosed by Hind.

Since Keane discloses a GUI programmed with Java and Hind discloses a GUI programmed with Java Server Page (JavaScript), the motivation for doing so would have been to allow an user interface with inline representation of steps in a multi-stepped process in conjunction with a network system to be programmed with an universal language like Java.

Therefore, it would have been obvious to combine Hind with Keane to obtain the invention as specified in the instant claim.

Final Office Action of April 25, 2008, pp. 6-7 (emphasis in original).

Claim 5 as amended is as follows:

5. The method of claim 2, wherein presenting the current step inline within the series of steps includes generating a response page using a JAVA Server Page.

B.I.i. The Proposed Combination, Considered as a Whole, Does Not Teach or Suggest all of the Claimed Features

The examiner failed to state a *prima facie* obviousness rejection against claim 5 because the proposed combination of references, considered as a whole, does not teach or suggest all of the features of claim 1, from which claim 5 depends. As shown above, *Keane* does not teach all of the features of claim 1. Furthermore, *Keane* does not suggest the claimed feature because *Keane* is devoid of disclosure regarding the features of claim 1. Additionally, *Keane* does not suggest the features of claim 1 because *Keane* is not concerned with identifying step components of steps because *Keane* is only concerned with printing batches of jobs.

Additionally, *Hind* does not teach or suggest all of the features of claim 1. *Hind* teaches a method for using Java Server Pages to enable transcoding of the content of a document requested by a client in order to tailor the output document according to application-specific characteristics. *Hind*, Abstract. *Hind* is completely devoid of disclosure regarding the features of claim 1.

Therefore, neither *Keane* nor *Hind* teach or suggest all of the features of claim 1. Accordingly, the proposed combination of references, considered as a whole, does not teach or suggest all of the features of claim 1. Hence, under the standards of *In re Royka*, the examiner failed to state a *prima facie* obviousness rejection against claim 1.

At least by virtue of the dependency of claim 5 on claim 1, the examiner failed to state a *prima facie* obviousness rejection against this grouping of claims. For similar reasons, the examiner failed to state a *prima facie* obviousness rejection against the remaining claims in this grouping of claims.

B.1.ii. The Examiner Failed To State a Proper Reason To Achieve the Legal Conclusion of Obviousness Under the Standards of *KSR Int'l*.

Additionally, the examiner failed to state a *prima facie* obviousness rejection against claim 5 because the examiner failed to state a proper reason to combine the references under the standards of *KSR Int'l*. Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006)).

The examiner did not state a proper reason to achieve the *legal conclusion* of obviousness under the standards of *KSR Int'l*. Instead, the examiner states:

The motivation for doing so would have been to allow an user interface with inline representation of steps in a multi-stepped process in conjunction with a network system to be programmed with an universal language like Java.

Final Office Action of April 25, 2008, p. 12.

However, the examiner only provided a purported advantage to combine the references. An advantage is not the legal conclusion of obviousness. For example, the examiner failed to connect that purported advantage to the *legal conclusion* of obviousness, are required by *KSR Int'l*. Under *KSR Int'l*., the examiner must provide some rational underpinning to the *legal conclusion* of

obviousness, not just state a purported advantage and then assume the legal conclusion of obviousness.

A rational underpinning for the legal conclusion of obviousness is not the same as an advantage; for example, one of ordinary skill would have to recognize the purported advantage, have a reason to implement the purported advantage, and also have no reason to avoid implementing the purported advantage in order to make the connection that one of ordinary skill would make the connection between the references in the first place. Additional logic would be required to state a compelling case for the *legal conclusion* of obviousness of the claim at issue; simply reciting an “advantage” is not enough. Therefore, under the standards of *KSR Int’l.*, the examiner failed to provide a rational underpinning to achieve the legal conclusion of obviousness. Hence, the examiner failed to state a *prima facie* obviousness rejection against claim 1 or any other claim in this grouping of claims.

B.1.iii. No Reason Exists To Combine the References Under the Standards of *KSR Int’l.*

Additionally, no rational reason exists to combine the references to achieve the invention of claim 5 when the references are considered as a whole. *Keane* is directed to batching print jobs. *Hind* is directed towards remote login to a graphical user interface server. The two references have nothing to do with *claim 1*, other than the two references require the use of a computer to perform the disclosed methods.

Because the references have nothing to do with each other or claim 5, one of ordinary skill could find no reason to combine the references to achieve the invention of claim 5, when the references are considered together as a whole. Accordingly, under the standards of *KSR Int’l.*, the examiner failed to state a *prima facie* obviousness rejection against claim 5. For similar reasons, the examiner failed to state a *prima facie* obviousness rejection against the remaining claims in this grouping of claims.

B.1.iv. *Hind* Is Non-Analogous Art.

The examiner has failed to state a *prima facie* obviousness rejection because *Hind* is non-analogous art. In order to rely on a reference as a basis for rejection, the reference must be either in the applicant’s field of endeavor or, if not, then reasonably pertinent to the particular problem with

which the inventor was concerned. *In re Oetiker*, 977 F.2d 1443, 24 U.S.P.Q.2d 1443, 1445 (Fed. Cir. 1992); *In re Deminski*, 796 F.2d 436, 442, 230 U.S.P.Q. 313, 315 (Fed. Cir. 1986).

In the case at hand, *Hind* is not in the same field of endeavor of Claim 5 and *Hind* is not reasonably pertinent to the particular problem with which Applicants were concerned. With regard to the first part of the test for analogous art, *Hind* is not in the same field of endeavor of Claim 5 because *Hind* is in the field of remote login to a graphical user interface server. In contrast, Claim 5 is in the field of presenting methods to users. The two fields are completely distinct from each other because the methods and techniques taught by the two references are completely distinct from each other. Thus, *Hind* fails the first test of *In re Oetiker*.

With regard to the second part of the test for analogous art, *Hind* is not reasonably pertinent to the particular problem with which Applicants were concerned. As established above, *Hind* is in the field of remote login to a graphical user interface server. Specifically, *Hind* is directed to the problem of providing Java server pages in transcoding environments. For example, *Hind* provides that:

In view of the advantageous aspects of using JSPs as discussed above, it would be desirable to incorporate JSPs in transcoding environments. There is currently no known technique for providing this capability. Accordingly, what is needed is a technique with which a JSP author can specify application-specific

characteristics to be used in the transcoding process, enabling the transcoding engine to generate output that is more precisely tailored for the requesting user.

Hind, col. 3, ll. 39-47.

In contrast, Claim 5 is directed to the problem of presenting methods to users. The problem addressed by *Hind* is completely distinct from the problem addressed by Claim 5. For this reason, *Hind* is not reasonably pertinent to the particular problem with which Applicants were concerned. Therefore, *Hind* fails the second part of the *In re Oetiker* test for analogous art.

Nevertheless, the examiner states that:

Keane and Hind are both analogous art because they are from the same field of endeavor of graphical user interface applications.

Final Office Action of April 25, 2008, p. 11.

The examiner appears to misunderstand or misapply the concept of non-analogous art. The question is not whether *Keane* and *Hind* are analogous to each other, but rather whether *Keane* and *Hind* are analogous to claim 5. As shown below, *Hind* is not analogous to claim 5.

In light of the requirements of *In re Oetiker* the examiner's characterizations of *Hind* and Claim 5 are too broad to establish that *Hind* is in the same field of endeavor as Claim 5. For example, the court in *In re Oetiker* stated that:

The examiner stated that "since garments commonly use hooks for securement", a person faced with the problem of unreliable maintenance of the pre-assembly configuration of an assembly line metal hose clamp would look to the garment industry art.

In re Oetiker, 977 F.2d 1443 at 1446.

The examiner in *In re Oetiker* attempted to use substantially the same argument as the present examiner. The argument is as follows: Because the reference and the claim both deal with the same broad class of problem, the reference is in the same field of endeavor as the claimed invention. However, the Court of Appeal for the Federal Circuit specifically states that this argument is incorrect:

It has not been shown that a person of ordinary skill, seeking to solve a problem of fastening a hose clamp, would reasonably be expected or motivated to look to fasteners for garments. The combination of elements from nonanalogous sources, in a manner that reconstructs the applicant's invention only with the benefit of *Hindsight*, is insufficient to present a *prima facie* case of obviousness. There must be some reason, suggestion, or motivation found in the prior art whereby a person of ordinary skill in the field of the invention would make the combination. That knowledge can not come from the applicant's invention itself.

Id. at 1447 (emphasis supplied).

The court ruled that the examiner failed to show that a person of ordinary skill solving a problem of fastening *hose clamps* would reasonably be expected or motivated to look to a reference dealing with *fasteners for garments*. Even though both technologies are in the same broad field of fastening objects, the reference was still considered to be non-analogous art.

In the case at hand, *Hind* is directed to the field of remote login to a graphical user interface server. In contrast, Claim 5 is directed to the field of presenting methods to users. These fields are more distinct from each other than the field of hose clamps and the field of fasteners for garments. In fact, the fields of hose clamps and fasteners for garments are *more* closely related because both

are fasteners. In stark contrast, the invention of Claim 5 is more divergent from the field of *Hind*. Therefore, under the standards of *In re Oetiker*, *Hind* is non-analogous art to Claim 5, notwithstanding the examiner's assertions to the contrary. Accordingly, the examiner failed to state a *prima facie* obviousness rejection against Claim 5.

The rejections of claims 6, 8-10, 15-16, 18-21, and 23 all rely on the combination of *Keane* and *Hind*. Therefore, because *Hind* is non-analogous art, the examiner also failed to state a *prima facie* obviousness rejection against this grouping of claims. Accordingly, this rejection is overcome.

B.1.v. Keane is Non-Analogous Art

Additionally, under the standards of *In re Oetiker*, *Keane* is also non-analogous art. As provided above, *Keane* is directed to managing print jobs. *Keane*, Title. Thus, *Keane* is not in the same field of invention as claim 5, which is presenting methods to users. Additionally, *Keane* is not reasonably related to the problem to be solved, which is overcoming shortcomings of help wizards. Instead, *Keane* is directed to the problem of managing different kind of user-initiated print jobs. The fact that *Keane* presents a method for drafting business cards is ancillary to this purpose, even if *Keane* taught all of the features of claim 1 (which it does not). Therefore, *Keane* fails both tests for analogous art set forth in *In re Oetiker*. Accordingly, *Keane* may not be relied upon to state a *prima facie* obviousness rejection against the claims. Hence, the examiner failed to state a *prima facie* obviousness rejection against this grouping of claims.

B.2. Refutation of Examiner's Response

In response to the facts established above, the Examiner asserts that:

Claims 5, 6, 8-10,15,16,18-21, and 23 are Rejected Under 35 USC - 103 (a):

Applicant's arguments filed for Claims 5, 6, 8-10, 15, 16, 18-21, and 23 have been fully considered but they are not persuasive. The applicant argues that *Keane* and *Hind* can not be combined because the examiner failed to disclose *prima facie* obviousness as a whole; failed to state a proper reason to achieve the legal conclusion of obviousness as stated by KSR because in the opinion of the applicant, an advantage is not the legal conclusion of obviousness, and *Hind* is not analogous art when combined with *Keane*.

Although the applicant presented a lengthy and an exhaustive argument, the applicant's reasoning is a bit short sighted. First, Keane and Hind can be combined to show *prima facie* because Keane discloses a GUI with Java programming and Hind discloses a GUI with a Java Server Page (Java Script). Java and Java Script are very similar except with one difference, Java requires compiling while Java Script does not. However, that difference does not take away from the *prima facie* because Java Script is derived from Java. Therefore, it is very well-known in the art that most programs that written in Java can also be written in Java Script, and vice versa. Second, the motivation is sufficient because both Keane and Hind disclose GUIs with Java related languages; moreover, combining these prior arts under reasons of obviousness is sufficient because the examiner used the requirements for 35 USC -1 03(a) rejection as recited in Graham vs. John Deere. Third, Hind and Keane are in the same field of endeavor because both Keane and Hind disclose GUIs with Java related languages.

Final Office Action of April 25, 2008, pp. 15-16.

However, the examiner is mistaken. *Hind* teaches a method for using Java Server Pages to enable transcoding of the content of a document requested by a client in order to tailor the output document according to application-specific characteristics. *Hind*, Abstract. *Hind* is completely devoid of disclosure regarding the features of claim 1.

C. GROUND OF REJECTION 3 (Claims 7, 17 and 22)

C.1. Response to Rejection

The second ground of rejection is the erroneous assertion that claims 7, 17 and 22 are obvious under 35 U.S.C. § 103 in view of *Keane*, *Hind*, and *Scheinblum*. This rejection is respectfully traversed. The examiner states that:

Keane and Hind disclose the limitations of claim 5

Keane discloses building a response page (Fig 1A and col 13, lines 59-65 → Keane discloses 'response page' in that the GUI communicates on a network with a communication protocol using HTML components. It is well-known in the art for the server to respond to a client using a page.

and the use of Java. (Col 13, lines 1-5 → Keane discloses a system which utilizes a Javascript.)

Keane and Hind does not appear to explicitly disclose *a method wherein building the response page using a Struts framework.*

However, Schein' discloses **a method presenting a Struts Framework**. (pg U-1, 1st paragraph → Schein' discloses a Struts Framework technology.)

Keane, Hind, and Schein' are analogous art because they are from the same field of endeavor of graphical user interface applications using Java.

At they time of the invention, it would have been obvious to one of ordinary skill in the art, having the teachings of Keane and Hind before him or her, to modify the GUI that identifies a current step within the series of steps to include communication with a client and server because it allows tasks to be listed and oriented in a network system with an universal web based language like Java, as disclosed by Keane, with a Java Server Page, as disclosed by Hind, and with Java based Struts framework, as disclosed by Schein',

Since Keane discloses a GUI programmed with Java and Hind discloses a GUI programmed with Java Server Page (JavaScript), the motivation for doing so would have been to allow templates of graphical interface user that presents with inline representation of steps in a multi-stepped process in conjunction with a network system to be programmed with an universal language like Java.

Therefore, it would have been obvious to combine Hind and Schein' with Keane to obtain the invention as specified in the instant claim.

Office action of April 25, 2008, pp. 12-14.

Claim 7 is as follows:

7. (Original) The method of claim 5, wherein generating a response page includes building the response page using a Struts framework.

C.I.i. The Proposed Combination of References, Considered as a Whole, Does Not Teach or Suggest all of the Claimed Features

The examiner failed to state a *prima facie* obviousness rejection against the claims because the proposed combination of the references, when considered as a whole, does not teach all of the features of the claims. To prove this fact, Applicants first address each reference in turn, and then the combination as a whole.

The rejected claims all depend from the independent claims. Therefore, at least by virtue of their dependency, *Keane* does not teach all of the features of the dependent claims, as asserted by

the examiner. Additionally, because *Keane* is devoid of disclosure regarding the above-described features of claim 1, *Keane* also does not suggest the features of claim 1.

Moreover, *Schienblum* does not teach or suggest the features of claim 1, and the examiner does not suggest otherwise. *Scheinblum* teaches how to create Java-based struts frameworks. *Scheinblum*, paragraph 1. *Scheinblum* is completely devoid of disclosure regarding the features of claim 1. Therefore, *Schienblum* also does not teach or suggest the features of claim 1.

Because neither *Keane* nor *Schienblum* teach or suggest the features of claim 1, the proposed combination of references, considered as a whole, does not teach or suggest the features of claim 1. Because the rejected claims all contain features similar to those presented in claim 1, the proposed combination of references, considered as a whole, does not teach or suggest the feature of the rejected dependent claims. Accordingly, the examiner failed to state a *prima facie* obviousness rejection against this grouping of claims.

C.1.ii. The Examiner Failed To State a Proper Reason To Achieve the Legal Conclusion of Obviousness Under the Standards of *KSR Int'l*.

As with the first obviousness rejection, the examiner only stated a purported advantage to combining the references. A purported advantage is not enough; instead, as announced in *KSR Int'l*., the examiner must provide a proper reason to achieve the *legal conclusion* of obviousness. A purported advantage may be part of the logical chain that establishes a reasoned analysis for the *legal conclusion*, but the examiner may not simply assume that the advantage alone compels the required analysis.

C.1.iii. *Keane* and *Schienblum* Are Non-Analogous to Claim 7.

Additionally, both *Keane* and *Schienblum* are non-analogous art to claim 7. *Keane* fails the tests of *In re Oetiker*, as described above.

Schienblum also fails the tests of *In re Oetiker* because *Schienblum* is not in the same field of endeavor as claim 7 and *Schienblum* is not reasonably related to the problem to be solved by claim 7. As provided above, *Schienblum* is directed to creation of Java-based frameworks, which has nothing to do with presenting steps in a process, as required in claim 7. Thus, *Schienblum* is non-analogous art under the standards of *In re Oetiker*.

Accordingly, *Schienblum* may not be relied upon to state a *prima facie* obviousness rejection against the claims. Hence, the examiner failed to state a *prima facie* obviousness rejection against this grouping of claims.

C.2. Refutation of Examiner's Response

In response to the facts established above, the Examiner asserts that:

Claims 7,17, and 22 are Rejected Under 35 USC - 103 (a):

Applicant's arguments filed for Claims 7, 17, and 22 have been fully considered but they are not persuasive. The applicant argues that Scheinblum can not be combined with Hind or Keane because Keane does not disclose the limitations of Claim I and Scheinblum is not the same field of endeavor.

Scheinblum discloses how to build a Java-based Struts framework. The examiner notes that Struts and Frames are GUI components and can be seen in Keane (Fig 4A and 48), where the GUI window is divided into frames. Since, Keane, Hind, and Scheinblum disclose GUIs and/or GUI components programmed by Java related languages, they are from the same field of endeavor. Moreover, Keane discloses the limitations of Claim I as stated above (see "Response to Arguments: Claims 1-4, 11-14, and 24 are Rejected Under 35 USC - 102 (b)").

Final Office Action of April 25, 2008, pp. 15-16.

However, the examiner is mistaken. The rejected claims all depend from the independent claims. Therefore, at least by virtue of their dependency, *Keane* does not teach all of the features of the dependent claims, as asserted by the examiner. Additionally, because *Keane* is devoid of disclosure regarding the above-described features of claim I, *Keane* also does not suggest the features of claim I.

D. CONCLUSION

As shown above, the Examiner has failed to state valid rejections against any of the claims. Therefore, Appellants request that the Board of Patent Appeals and Interferences reverse the rejections. Additionally, Appellants request that the Board direct the examiner to allow the claims.

/Theodore D. Fay/

Reg. No. 48,504
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777

TDF/ngf

CLAIMS APPENDIX

The text of the claims involved in the appeal is as follows:

1. A method for presenting a step of a task, wherein the task includes a series of steps to be performed, the method comprising:

 identifying a current step within the series of steps;

 retrieving a step component for the current step; and

 presenting the current step inline within the series of steps such that the step component is presented in context within the series of steps.
2. The method of claim 1, wherein identifying a current step within the series of steps includes receiving a request from a client.
3. The method of claim 2, wherein the request from the client identifies a user selection of the current step within the series of steps.
4. The method of claim 2, wherein the request is a HyperText Transfer Protocol request.
5. The method of claim 2, wherein presenting the current step inline within the series of steps includes generating a response page using a JAVA Server Page.

6. The method of claim 5, wherein retrieving a step component for the current step includes retrieving the step component using a Tiles framework.
7. The method of claim 5, wherein generating a response page includes building the response page using a Struts framework.
8. The method of claim 5, wherein the response page includes a navigation tile, wherein the navigation tile presents a plurality of tasks.
9. The method of claim 8, further comprising:
 - identifying a current task within the plurality of tasks;
 - retrieving a task tile for the current task; and
 - presenting the task tile as a series of steps.
10. The method of claim 5, wherein presenting the current step inline within the series of steps further includes sending the response page to the client.
11. An apparatus for presenting a step of a task, wherein the task includes a series of steps to be performed, the apparatus comprising:
 - means for identifying a current step within the series of steps;
 - means for retrieving a step component for the current step; and
 - means for presenting the current step inline within the series of steps such that the step component is presented in context within the series of steps.

12. The apparatus of claim 11, wherein the means for identifying a current step within the series of steps includes means for receiving a request from a client.
13. The apparatus of claim 12, wherein the request from the client identifies a user selection of the current step within the series of steps.
14. The apparatus of claim 12, wherein the request is a HyperText Transfer Protocol request.
15. The apparatus of claim 12, wherein the means for presenting the current step inline within the series of steps includes means for generating a response page using a JAVA Server Page.
16. The apparatus of claim 15, wherein the means for retrieving a step component for the current step includes means for retrieving the step component using a Tiles framework.
17. The apparatus of claim 15, wherein the means for generating a response page includes means for building the response page using a Struts framework.
18. The apparatus of claim 15, wherein the response page includes a navigation tile, wherein the navigation tile presents a plurality of tasks.
19. The apparatus of claim 18, further comprising:
means for identifying a current task within the plurality of tasks;

means for retrieving a task tile for the current task; and

means for presenting the task tile as a series of steps.

20. The apparatus of claim 15, wherein the means for presenting the current step inline within the series of steps further includes means for sending the response page to the client.

21. A server for presenting a step of a task, wherein the task includes a series of steps to be performed, the method comprising:

a controller, wherein the controller receives a request from a client, wherein the request identifies a current step within the series of steps, and wherein the controller retrieves a step component for the current step; and

a JAVA Server Page, wherein the JAVA Server Page builds a response page and wherein the response page presents the current step inline within the series of steps such that the step component is presented in context within the series of steps.

22. The server of claim 21, wherein the controller and the JAVA Server Page use a Struts framework.

23. The server of claim 21, wherein the controller and the JAVA Server Page use a Tiles framework.

24. A computer program product comprising:
- a recordable-type medium having instructions for presenting a step of a task, wherein the task includes a series of steps to be performed, the computer program product comprising:
- instructions for identifying a current step within the series of steps;
- instructions for retrieving a step component for the current step; and
- instructions for presenting the current step inline within the series of steps such that the step component is presented in context within the series of steps.

EVIDENCE APPENDIX

This appeal brief presents no additional evidence.

RELATED PROCEEDINGS APPENDIX

This appeal has no related proceedings.